

862.C1922



PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

2756

#2  
SSK

102500

In re Application of:

NOBORU HAMADA

Appln. No.: 09/588,672

Filed: June 6, 2000

For: NETWORK DEVICE  
MANAGING APPARATUS  
AND METHOD

Examiner: Not Yet Known

Group Art Unit: 2756

October 19, 2000

Commissioner for Patents  
Washington, D.C. 20231

CLAIM TO PRIORITY

TECH CENTER 2700

OCT 24 2000

RECEIVED

Sir:

Applicant hereby claims priority under the  
International Convention and all rights to which he is  
entitled under 35 U.S.C. § 119 based upon the following  
Japanese Priority Application:

11-165569 filed on June 11, 1999

A certified copy of the priority document, together  
with an English translation of the first page of the same,  
containing the filing data, is enclosed.

CFM 1922 US  
09/588,672

日 本 国 特 許 庁  
PATENT OFFICE  
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されて  
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed  
with this Office.

出 願 年 月 日  
Date of Application: 1999年 6月11日

出 願 番 号  
Application Number: 平成11年特許願第165569号

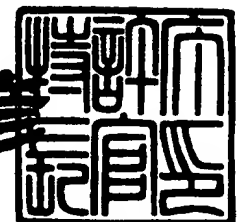
出 願 人  
Applicant(s): キヤノン株式会社

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2000年 6月29日

特許庁長官  
Commissioner,  
Patent Office

近 藤 隆 彦



出証番号 出証特2000-3050745

【書類名】 特許願

【整理番号】 3933107

【提出日】 平成11年 6月11日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 15/00

【発明の名称】 ネットワークデバイス管理装置及び方法

【請求項の数】 10

【発明者】

    【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社  
社内

    【氏名】 浜田 昇

【特許出願人】

    【識別番号】 000001007

    【氏名又は名称】 キヤノン株式会社

【代理人】

    【識別番号】 100076428

    【弁理士】

    【氏名又は名称】 大塚 康德

    【電話番号】 03-5276-3241

【選任した代理人】

    【識別番号】 100093908

    【弁理士】

    【氏名又は名称】 松本 研一

    【電話番号】 03-5276-3241

【選任した代理人】

    【識別番号】 100101306

    【弁理士】

    【氏名又は名称】 丸山 幸雄

    【電話番号】 03-5276-3241

【手数料の表示】

【予納台帳番号】 003458

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9704672

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ネットワークデバイス管理装置及び方法

【特許請求の範囲】

【請求項 1】 起動後に少なくとも 1 度はネットワーク管理用パケットをブロードキャストするデバイスが接続されたネットワークの管理装置であって、

パケットを受信するパケット受信手段と、

前記パケット受信手段で受信したパケットが、ネットワーク管理用のパケットであるか判定するパケット判定手段と、

前記パケット判定手段でネットワーク管理用パケットであると判断された場合、そのパケットから、そのパケットを送信したデバイスのアドレスを取得するデバイスアドレス取得手段と、

前記デバイスアドレスを登録するデバイスアドレス登録手段と  
を備えることを特徴とするネットワークデバイス管理装置。

【請求項 2】 起動後に少なくとも 1 度はネットワーク管理用パケットをブロードキャストするデバイスが接続されたネットワークの管理装置であって、

パケットを受信するパケット受信手段と、

前記パケット受信手段で受信したパケットが、ネットワーク管理用のパケットであるか判定するパケット判定手段と、

前記パケット判定手段でネットワーク管理用パケットであると判断された場合、そのパケットを送信したデバイスに対して、そのデバイスが監視の対象となるデバイスであることを確認するための確認パケットを送信する送信手段と、

前記監視パケットに対する応答に基づいて、その応答を返したデバイスの状態をアドレスと共に登録するデバイスアドレス登録手段と  
を備えることを特徴とするネットワークデバイス管理装置。

【請求項 3】 前記応答は、その応答元のデバイスが管理対象のデバイスであれば正常応答であり、該デバイスの稼働状態を示す情報を含むことを特徴とする請求項 2 に記載のネットワークデバイス管理装置。

【請求項 4】 前記送信手段は、プリンタの状態を取得するための確認パケットを送信し、前記アドレス登録手段は、確認パケットに対する応答が正常応答

であれば、応答元のデバイスのアドレスと共に、そのデバイスがプリンタであることを登録することを特徴とする請求項 2 に記載のネットワークデバイス管理装置。

【請求項 5】 起動後に少なくとも 1 度はネットワーク管理用パケットをブロードキャストするデバイスが接続されたネットワークの管理方法であって、  
パケットを受信するパケット受信工程と、

前記パケット受信工程で受信したパケットが、ネットワーク管理用のパケットであるか判定するパケット判定工程と、

前記パケット判定工程でネットワーク管理用パケットであると判断された場合、そのパケットから、そのパケットを送信したデバイスのアドレスを取得するデバイスアドレス取得工程と、

前記デバイスアドレスを登録するデバイスアドレス登録工程と  
を備えることを特徴とするネットワークデバイス管理方法。

【請求項 6】 起動後に少なくとも 1 度はネットワーク管理用パケットをブロードキャストするデバイスが接続されたネットワークの管理方法であって、  
パケットを受信するパケット受信工程と、

前記パケット受信工程で受信したパケットが、ネットワーク管理用のパケットであるか判定するパケット判定工程と、

前記パケット判定工程でネットワーク管理用パケットであると判断された場合、そのパケットを送信したデバイスに対して、そのデバイスが監視の対象となるデバイスであることを確認するための確認パケットを送信する送信工程と、

前記監視パケットに対する応答に基づいて、その応答を返したデバイスの状態をアドレスと共に登録するデバイスアドレス登録工程と  
を備えることを特徴とするネットワークデバイス管理方法。

【請求項 7】 前記応答は、その応答元のデバイスが管理対象のデバイスであれば正常応答であり、該デバイスの稼働状態を示す情報を含むことを特徴とする請求項 6 に記載のネットワークデバイス管理方法。

【請求項 8】 前記送信工程は、プリンタの状態を取得するための確認パケットを送信し、前記アドレス登録工程は、確認パケットに対する応答が正常応答

であれば、応答元のデバイスのアドレスと共に、そのデバイスがプリンタであることを登録することを特徴とする請求項 6 に記載のネットワークデバイス管理方法。

【請求項 9】 コンピュータにより、  
パケットを受信するパケット受信手段と、  
前記パケット受信手段で受信したパケットが、ネットワーク管理用のパケットであるか判定するパケット判定手段と、  
前記パケット判定手段でネットワーク管理用パケットであると判断された場合、そのパケットから、そのパケットを送信したデバイスのアドレスを取得するデバイスアドレス取得手段と、  
前記デバイスアドレスを登録するデバイスアドレス登録手段と  
を実現するためのコンピュータプログラムを格納するコンピュータ可読の記憶媒体。

【請求項 10】 コンピュータにより、  
パケットを受信するパケット受信手段と、  
前記パケット受信手段で受信したパケットが、ネットワーク管理用のパケットであるか判定するパケット判定手段と、  
前記パケット判定手段でネットワーク管理用パケットであると判断された場合、そのパケットを送信したデバイスに対して、そのデバイスが監視の対象となるデバイスであることを確認するための確認パケットを送信する送信手段と、  
前記監視パケットに対する応答に基づいて、その応答を返したデバイスの状態をアドレスと共に登録するデバイスアドレス登録手段と  
を実現するためのコンピュータプログラムを格納するコンピュータ可読の記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、例えばコンピュータネットワークに接続されたデバイス等を管理するためのネットワークデバイス管理装置及び方法に関する。

## 【0002】

## 【従来の技術】

近年、コンピュータを相互に接続したローカルエリアネットワーク（LAN）が普及しており、このようなローカルエリアネットワークは、ビルのフロアまたはビル全体、ビル群（構内）、地域、あるいはさらに大きいエリアにわたって構築することができる。このようなネットワークは更に相互接続され、世界的規模のネットワークにも接続することができる。このような相互接続されたそれぞれのLANは、多様なハードウェア相互接続技術といくつかのネットワークプロトコルを持つことがある。

## 【0003】

他と切り離された簡単なLANは個々のユーザが管理することができる。すなわち、ユーザが機器を取り替えたり、ソフトウェアをインストールしたり、問題点を診断したりすることができる。

## 【0004】

しかし一方、規模の大きい複雑なLANや相互接続された大きなLANグループは「管理」を必要とする。「管理」とは、人間のネットワーク管理者とその管理者が使用するソフトウェアの両方による管理を意味する。本願においては、「管理」とはシステム全体を管理するためのソフトウェアによる管理を意味し、「ユーザ」とはネットワーク管理ソフトウェアを使用する人を意味するものとする。このユーザは、通常、システム管理責任者である。ユーザは、ネットワーク管理ソフトウェアを使うことによって、ネットワーク上で管理データを得て、このデータを変更することができる。

## 【0005】

大規模ネットワークシステムは、通常、機器の増設と除去、ソフトウェアの更新、及び問題の検出などを絶えず行うことが必要な動的システムである。一般に、様々な人が所有する、様々な業者から供給される様々なシステムが存在する。

## 【0006】

このような大規模ネットワークシステムを構成するネットワーク上のデバイスを管理するための方法として、これまでにいくつかの試みが数多くの標準機関で



なされている。国際標準化機構（I S O）は開放型システム間相互接続（Open S ystem Interconnection, O S I）モデルと呼ばれる汎用基準フレームワークを提供した。ネットワーク管理プロトコルのO S Iモデルは、共通管理情報プロトコル（Comon Management Information Protocol, C M I P）と呼ばれる。C M I Pはヨーロッパの共通ネットワーク管理プロトコルである。

【 0 0 0 7 】

また近年では、より共通性の高いネットワーク管理プロトコルとして、簡易ネットワーク管理プロトコル（Simple Network Management Protocol, S N M P）と呼ばれるC M I Pに関連する一変種のプロトコルがある。（「T C P / I P ネットワーク管理入門実用的な管理をめざして」M. T. ローズ＝著／西田竹志＝訳（株）トッパン発行1 9 9 2年8月2 0日初版を参照。）

このS N M Pネットワーク管理技術によれば、ネットワーク管理システムには、少なくとも1つのネットワーク管理ステーション（N M S）、各々がエージェントを含むいくつかの管理対象ノード、及び管理ステーションやエージェントが管理情報を交換するために使用するネットワーク管理プロトコルが含まれる。ユーザは、N M S上でネットワーク管理ソフトウェアを用いて管理対象ノード上のエージェントソフトウェアと通信することにより、ネットワーク上のデータを得、またデータを変更することができる。

【 0 0 0 8 】

ここでエージェントとは、各々のターゲット装置についてのバックラウンドプロセスとして走るソフトウェアである。ユーザがネットワーク上の装置に対して管理データを要求すると、管理ソフトウェアはオブジェクト識別情報を管理パケットまたはフレームに入れてターゲットエージェントへ送り出す。エージェントは、そのオブジェクト識別情報を解釈して、そのオブジェクト識別情報に対応するデータを取り出し、そのデータをパケットに入れてユーザに送り返す。時には、データを取り出すために対応するプロセスが呼び出される場合もある。

【 0 0 0 9 】

またエージェントは、自分の状態に関するデータをデータベースの形式で保持している。このデータベースのことを、M I B (Management Information Base)

と呼ぶ。図4は、M I Bの構造を示す概念図である。図4に示すように、M I Bは木構造のデータ構造をしており、全てのノードが一意に番号付けされている。図4において、かっこ内に書かれている番号が、そのノードの識別子である。例えば、図4においてノード4 0 1の識別子は1である。ノード4 0 2の識別子は、ノード4 0 1の下の子なので、1・3と表記される。同様に、ノード4 0 3の識別子は、1・3・6・1・2と表記される。このノードの識別子のことを、オブジェクト識別子(OBJECT IDENTIFIER)と呼ぶ。

#### 【0 0 1 0】

このM I Bの構造は、管理情報構造 (S M I : Structure of Management Information) と呼ばれ、RFC1155 Structure and Identification of Management Information for TCP/IP-based Internetsで規定されている。

#### 【0 0 1 1】

図4には、標準として規定されているM I Bのうち、一部のもののみを抜き出して記載してある。

#### 【0 0 1 2】

次に、SNMPプロトコルについて簡単に説明する。ネットワーク管理ユーティリティソフトウェアが動作しているPC（以下、マネージャと呼称する）とSNMPエージェントが動作している管理対象ネットワークデバイス（以下、エージェントと呼称する）とは、SNMPプロトコルを用いて通信を行う。SNMPプロトコルには5種類のコマンドがあり、それぞれGet-request、Get-next-request、Get-response、Set-request、Trapと呼ばれる。これらのコマンドがマネージャとエージェントの間でやりとりされる様子を図8に示す。

#### 【0 0 1 3】

Get-requestおよびGet-next-requestは、マネージャがエージェントのM I Bオブジェクトの値を取得するために、マネージャがエージェントに対して送出するコマンドである。このコマンドを受け取ったエージェントは、M I Bの値をマネージャに通知するために、マネージャに対してGet-responseコマンドを送出する（8 0 1および8 0 2）。

## 【0014】

Set-request は、マネージャがエージェントのMIBオブジェクトの値を設定するために、マネージャがエージェントに対して送出するコマンドである。このコマンドを受け取ったエージェントは、設定結果をマネージャに通知するために、マネージャに対してGet-response コマンドを送出する(803)。

## 【0015】

Trap は、エージェントが自分自身の状態の変化をマネージャに対して通知するために、エージェントがマネージャに対して送出するコマンドである(804)。

## 【0016】

図7に、Trap 以外のコマンド、即ちGet-request、Get-next-request、Get-response およびSet-request のフォーマットを示す。

## 【0017】

700 は、SNMP メッセージを示す。SNMP メッセージは、バージョン701、コミュニティ名702 およびPDU と呼ばれる領域703 からなる。PDU 703 を詳細に示したものが710 である。PDU 710 は、PDU タイプ711、リクエストID 712、エラーステータス713、エラーインデックス714 およびMIB 情報715 からなる。PDU タイプ711 には、コマンドを識別する値が格納される。即ちこのフィールドの値が0 であるならばGet-request、1 であるならばGet-next-request、2 であるならばGet-response、3 であるならばSet-request と識別される。また、エラーステータス713 には、エラー情報を示す値が格納される。エラーが無い場合には、このフィールドの値は0 である。また、MIB 情報715 には、オブジェクトID とその値が組みになって格納される。

## 【0018】

次に、管理が必要な大規模なネットワークについて説明する。

## 【0019】

図1は、プリンタをネットワークに接続するためのネットワークボード（NB）101を、開放型アーキテクチャを持つプリンター102へつなげた場合を示す図である。NB101はローカルエリアネットワーク（LAN）100へ、例えば、同軸コネクタをもつEthernetインターフェース10Base-2や、RJ-45を持つ10Base-T等のLANインターフェースを介してつながれている。

#### 【0020】

PC103やPC104等の複数のパーソナルコンピューター（PC）もまた、100に接続されており、ネットワークオペレーティングシステムの制御の下、これらのPCはNB101と通信することができる。PCの一つ、例えばPC103を、ネットワーク管理部として使用するよう指定することができる。PCに、PC104に接続されているプリンター105のようなプリンターを接続してもよい。

#### 【0021】

また、LAN100にファイルサーバー106が接続されており、これは大容量（例えば100億バイト）のネットワークディスク107に記憶されたファイルへのアクセスを管理する。プリントサーバー108は、接続されたプリンター109a及び109b、又は遠隔地にあるプリンター105などのプリンターに印刷を行わせる。また他の図示しない周辺機器をLAN100に接続してもよい。

#### 【0022】

更に詳しくは、図1に示すネットワークは、様々なネットワークメンバー間で効率良く通信を行うために、NovellやUNIXのソフトウェアなどのネットワークソフトウェアを使用することができる。どのネットワークソフトウェアを使用することも可能であるが、例えば、Novell社のNetWare（Novell社の商標。以下省略）ソフトウェアを使用することができる。このソフトウェアパッケージに関する詳細な説明は、NetWareパッケージに同梱されているオンラインドキュメンテーションを参照のこと。これは、Novell社からNetWareパッケージとともに購入可能である。

#### 【0023】

図1の構成について簡潔に説明すると、ファイルサーバー106は、LANメンバー間でデータのファイルの受信や、記憶、キューイング、キャッシング、及び送信を行う、ファイル管理部としての役割を果たす。例えば、PC103及びPC104それぞれによって作られたデータファイルは、ファイルサーバー106へ送られ、ファイルサーバー106はこれらのデータファイルを順に並べ、そしてプリントサーバー108からのコマンドに従って、並べられたデータファイルをプリンター109aへ送信する。

#### 【0024】

またPC103とPC104はそれぞれ、データファイルの生成や、生成したデータファイルのLAN100への送信や、また、LAN100からのファイルの受信や、更にそのようなファイルの表示及び／又は処理を行うことのできる、通常のPCで構成される。図1にパーソナルコンピュータ機器が示されているが、ネットワークソフトウェアを実行するのに適切であるような、他のコンピュータ機器を含んでもよい。例えば、UNIXのソフトウェアを使用している場合に、UNIXワークステーションをネットワークに含んでもよく、これらのワークステーションは、適切な状況下で、図示されているPCと共に使用することができる。

#### 【0025】

通常、LAN100などのLANは、一つの建物内の一つの階又は連続した複数の階でのユーザーグループ等の、幾分ローカルなユーザーグループにサービスを提供する。例えば、ユーザーが他の建物や他県に居るなど、あるユーザーが他のユーザーから離れるに従って、ワイドエリアネットワーク(WAN)を作ってもよい。WANは、基本的には、いくつかのLANを高速度サービス総合デジタルネットワーク(ISDN)電話線等の高速度デジタルラインで接続して形成された集合体である。従って、図1に示すように、LAN100と、LAN110と、LAN120とは変調／復調(MODEM)／トランスポンダー130及びバックボーン140を介して接続されWANを形成する。これらの接続は、数本のバスによる単純な電氣的接続である。それぞれのLANは専用のPCを含み、また、必ずしも必要なわけではないが、通常はファイルサーバー及びプリントサ

ーバーを含む。

【0026】

従って図1に示すように、LAN110は、PC111と、PC112と、ファイルサーバー113と、ネットワークディスク114と、プリントサーバー115と、プリンター116及びプリンター117を含む。対照的に、LAN120はPC121とPC122のみを含む。LAN100と、LAN110と、LAN120とに接続されている機器は、WAN接続を介して、他のLANの機器の機能にアクセスすることができる。

【0027】

エージェントの実装例として、プリンタをネットワークに接続するためのネットワークボード上にエージェントを実装することが考えられる。これにより、プリンタをネットワーク管理ソフトウェアによる管理の対象とすることができる。ユーザは、ネットワーク管理ソフトウェアを用いて制御対象のプリンタの情報を得、また状態を変更することができる。より具体的には、例えばプリンタの液晶ディスプレイに表示されている文字列を取得したり、デフォルトの給紙カセットを変更したりすることができる。以下、エージェントを実装したネットワークボード(NB)をプリンタに接続する実施形態について説明する。図2に示すように、好ましくは、NB101は、プリンター102の内部拡張I/Oスロットに内蔵されており、NB101は、下に示す処理及びデータ記憶機能を持つ「埋め込まれた」ネットワークノードとなる。このNB101の構成により、大きなマルチエリアWANネットワークを統括及び管理するための、特徴的な補助機能を持つという利点をもたらす。これらの補助機能は、例えば、ネットワーク上の遠隔地(ネットワーク統括者の事務所など)からのプリンター制御及び状態観察や、各印刷ジョブ後の次のユーザーのための保証初期環境を提供するためのプリンター構成の自動管理、及びプリンターの負荷量を特徴付け、あるいはトナーカートリッジの交換スケジュールを組むためにネットワークを通してアクセスできる、プリンターログ又は使用統計を含む。

【0028】

このNB設計において重要な要因は、共有メモリ等の両方向インターフェース

を介して、NB 1 0 1 からプリンター制御状態にアクセスする機能である。共有メモリ以外に、S C S I インターフェース等のインターフェースを使用することもできる。これにより、多数の便利な補助機能のプログラムができるように、プリンター操作情報をNB 1 0 1 又は外部ネットワークノードへ送出することができる。印刷画像データ及び制御情報のブロックは、NB 1 0 1 上にあるマイクロプロセッサによって構成され、共有メモリに記述され、そして、プリンター 1 0 2 によって読み込まれる。同様に、プリンター状態情報は、プリンター 1 0 2 から共有メモリへ送られ、そこからNBプロセッサによって読み込まれる。

## 【 0 0 2 9 】

図 2 は、NB 1 0 1 をプリンター 1 0 2 にインストールした状態を示す断面図である。図 2 に示すように、NB 1 0 1 はネットワーク接続の為のフェースプレート 1 0 1 b を設置した印刷回路ボード 1 0 1 a から構成されており、コネクタ 1 7 0 を介してプリンターインターフェースやード 1 5 0 に接続されている。プリンターインターフェースカード 1 5 0 は、プリンター 1 0 2 のプリンターエンジンを直接制御する。印刷データ及びプリンター状態コマンドは、NB 1 0 1 からコネクタ 1 7 0 を介して、プリンターインターフェースカード 1 5 0 へ入力され、また、プリンター状態情報はプリンターインターフェースカード 1 5 0 からやはりコネクタ 1 7 0 を介して得られる。NB 1 0 1 はこの情報を、フェースプレート 1 0 1 b のネットワークコネクタを介して、LAN 1 0 0 上で通信する。同時に、プリンター 1 0 2 は、従来のシリアルポート 1 0 2 a 及びパラレルポート 1 0 2 b から、印刷データを受信することもできる。

## 【 0 0 3 0 】

図 3 は、NB 1 0 1 とプリンター 1 0 2 と LAN 1 0 0 との電氣的接続を示すブロック図である。NB 1 0 1 は、LAN 1 0 0 へは LAN インターフェースを介して、プリンター 1 0 2 へはプリンターインターフェースカード 1 5 0 を介して直接接続されている。NB 1 0 1 上には NB 1 0 1 を制御するためのマイクロプロセッサ 3 0 1 と、マイクロプロセッサ 3 0 1 の動作プログラムを格納するための ROM 3 0 3 と、マイクロプロセッサ 3 0 1 がプログラムを実行する上でワークとして用いるための RAM 3 0 2 と、NB 1 0 1 とプリンタインタフ

エースカード 1 5 0 とが相互にデータをやりとりするための共有メモリ 2 0 0 が  
あり、内部バスを通じて相互に接続されている。NB 1 0 1 が SNMP のエー  
ジェントとして動作するためのプログラムは ROM 3 0 3 に格納されている。マイ  
クロプロセッサ 3 0 1 は、ROM 3 0 3 に格納されたプログラムに従って動作し  
、ワークエリアとして RAM 3 0 2 を用いる。また、プリンターインターフェー  
スカード 1 5 0 と相互に通信するためのバッファ領域として共有メモリ 2 0 0 を  
用いる。

【 0 0 3 1 】

プリンターインタフェースカード 1 5 0 上のマイクロプロセッサ 1 5 1 は N  
B 1 0 1 とのデータのアクセスを、NB 1 0 1 に設置されている共有メモリ 2 0  
0 を介して行う。プリンターインタフェースカード 1 5 0 上のマイクロプロセッ  
サ 1 5 1 は、実際に印刷機構を動かすプリンターエンジン 1 6 0 とも通信する

【 0 0 3 2 】

一方、ネットワーク管理ソフトウェアが稼動する PC 側について、以下で説明  
する。

【 0 0 3 3 】

図 5 は、ネットワーク管理ソフトウェアが稼動可能な PC の構成を示すブロッ  
ク図である。

【 0 0 3 4 】

図 5 において、5 0 0 は、ネットワーク管理ソフトウェアが稼動する PC であ  
り、図 1 における 1 0 3 と同等である。PC 5 0 0 は、ROM 5 0 2 もしくはハ  
ードディスク (HD) 5 1 1 に記憶された、あるいはフロッピーディスクドライ  
ブ (FD) 5 1 2 より供給されるネットワーク管理プログラムを実行する CPU  
5 0 1 を備え、システムバス 5 0 4 に接続される各デバイスを総括的に制御する

【 0 0 3 5 】

5 0 3 は RAM で、CPU 5 0 1 の主メモリ、ワークエリア等として機能する



## 【 0 0 3 6 】

5 0 5 はキーボードコントローラ ( K B C ) で、キーボード ( K B ) 5 0 9 や不図示のポインティングデバイス等からの指示入力を制御する。5 0 6 は C R T コントローラ ( C R T C ) で、C R T ディスプレイ ( C R T ) 5 1 0 の表示を制御する。5 0 7 はディスクコントローラ ( D K C ) で、ブートプログラム、種々のアプリケーション、編集ファイル、ユーザファイルそしてネットワーク管理プログラム等を記憶するハードディスク ( H D ) 5 1 1 およびフロッピーディスクコントローラ ( F D ) 5 1 2 とのアクセスを制御する。5 0 8 はネットワークインタフェースカード ( N I C ) で、L A N 1 0 0 を介して、エージェントあるいはネットワーク機器と双方向にデータをやりとりする。

## 【 0 0 3 7 】

次に、従来例におけるネットワーク管理ソフトウェアの構成について説明する。

## 【 0 0 3 8 】

従来例におけるネットワーク管理装置は、図 5 に示したようなネットワーク管理装置を実現可能な P C と同様の構成の P C 上に実現される。ハードディスク ( H D ) 5 1 1 には、後述のすべての説明で動作主体となる本願に係るネットワーク管理ソフトウェアのプログラムが格納される。後述のすべての説明において、特に断りのない限り、実行の主体はハード上は C P U 5 0 1 である。一方、ソフトウェア上の制御の主体は、ハードディスク ( H D ) 5 1 1 に格納されたネットワーク管理ソフトウェアである。本従来例においては、O S は例えば、ウィンドウズ 9 5 ( マイクロソフト社製 ) を想定しているが、これに限るものではない。

## 【 0 0 3 9 】

なお本願に係るネットワーク管理プログラムは、フロッピーディスクや C D - R O M などの記憶媒体に格納された形で供給されても良く、その場合には図 5 に示すフロッピーディスクコントローラ ( F D ) 5 1 2 または不図示の C D - R O M ドライブなどによって記憶媒体からプログラムが読み取られ、ハードディスク ( H D ) 5 1 1 にインストールされる。

## 【 0 0 4 0 】

図 6 は、本従来例に係るネットワーク管理ソフトウェアのモジュール構成図である。このネットワーク管理ソフトウェアは、図 5 におけるハードディスク 5 1 1 に格納されており、CPU 5 0 1 によって実行される。その際、CPU 5 0 1 はワークエリアとして RAM 5 0 3 を使用する。

#### 【0 0 4 1】

図 6 において、6 0 1 はデバイスリストモジュールと呼ばれ、ネットワークに接続されたデバイスを一覧にして表示するモジュールである。（一覧表示の様子については、後ほど図 1 5 を用いて説明する。）6 0 2 は全体制御モジュールと呼ばれ、デバイスリストからの指示をもとに、他のモジュールを統括する。6 0 3 はコンフィグレータと呼ばれ、エージェントのネットワーク設定に関する特別な処理を行うモジュールである。6 0 4 は、探索モジュールと呼ばれ、ネットワークに接続されているデバイスを探索するモジュールである。探索モジュール 6 0 4 によって探索されたデバイスが、デバイスリスト 6 0 1 によって一覧表示される。6 0 5 は、プリントジョブの状況を NetWare API 6 1 6 を用いてネットワークサーバから取得する NetWare ジョブモジュールである。（なお、NetWare API については、例えば Novell 社から発行されている "NetWare Programmer's Guide for C" 等を参照のこと。この書籍はノベル株式会社から購入可能である。）6 0 6 および 6 0 7 は後述するデバイス詳細ウィンドウを表示するための UI (User Interface) モジュールであり、詳細情報を表示する対象機種毎に UI モジュールが存在する。6 0 8 および 6 0 9 は制御モジュールと呼ばれ、詳細情報を取得する対象機種に特有の制御を受け持つモジュールである。UI モジュールと同様に、制御モジュールも詳細情報を表示する対象機種毎に存在する。制御 A モジュール 6 0 8 および制御 B モジュール 6 0 9 は、MIB モジュール 6 1 0 を用いて管理対象デバイスから MIB データを取得し、必要に応じてデータの変換を行い、各々対応する UI A モジュール 6 0 6 または UI B モジュール 6 0 7 にデータを渡す。

#### 【0 0 4 2】

さて、MIB モジュール 6 1 0 は、オブジェクト識別子とオブジェクトキーとの変換を行うモジュールである。ここでオブジェクトキーとは、オブジェクト識

別子と一対一に対応する 3 2 ビットの整数のことである。オブジェクト識別子は可変長の識別子であり、ネットワーク管理ソフトウェアを実装する上で扱いが面倒なので、本願に係るネットワーク管理ソフトウェアにおいてはオブジェクト識別子と一対一に対応する固定長の識別子を内部的に用いている。M I B モジュール 6 1 0 より上位のモジュールはこのオブジェクトキーを用いて M I B の情報を扱う。これにより、ネットワーク管理ソフトウェアの実装が楽になる。

#### 【 0 0 4 3 】

6 1 1 は S N M P モジュールと呼ばれ、S N M P パケットの送信と受信を行う。

#### 【 0 0 4 4 】

6 1 2 は共通トランスポートモジュールと呼ばれ、S N M P データを運搬するための下位プロトコルの差を吸収するモジュールである。実際には、動作時にユーザが選択したプロトコルによって、I P X ハンドラ 6 1 3 か U D P ハンドラ 6 1 4 のいずれかがデータを転送する役割を担う。なお、U D P ハンドラは、実装として WinSock 6 1 7 を用いている。(WinSock については、例えば Windows socket API v1.1 の仕様書を参照のこと。このドキュメントは、複数箇所から入手可能であるが、例えばマイクロソフト社製のコンパイラである Visual C++ に同梱されている。)

コンフィグレータ 6 0 3 が用いる現在のプロトコル 6 1 5 というのは、動作時にユーザが選択している I P X プロトコルか U D P プロトコルのいずれかのことを示す。

#### 【 0 0 4 5 】

本従来例で使用する探索モジュール 6 0 4 と M I B モジュール 6 1 0 との間のインタフェースについて説明する。

#### 【 0 0 4 6 】

M I B モジュール 6 1 0 は図 9 に示す C 言語の A P I (Application Program Interface) を上位モジュールに提供する。

#### 【 0 0 4 7 】

最初に、上位モジュールは M I B モジュールとの間で、指定したアドレスに対

するインタフェース（これをポートと呼ぶ）を開設するために、MIBOpen API 9 0 1 呼び出しを行う。MIBモジュールは、開設されたインタフェースを識別するための識別子（これをポート識別子と呼ぶ）を上位モジュールに返す（MIBOpen API 9 0 1 の第一引数portに返される値）。以降上位モジュールはポート識別子を用いてMIBモジュールとのやりとりを行う。

#### 【0048】

ここで指定するアドレスは、動作しているプロトコルのアドレスであり、IPプロトコルの場合はIPアドレス、NetWareプロトコルの場合はNetWareアドレスである。さらにブロードキャストアドレスを指定することもできる。

#### 【0049】

ブロードキャストアドレスを指定してポートをオープンした場合は、ブロードキャストアドレスに応答する複数のデバイスと通信を行うことが可能である。

#### 【0050】

上位モジュールはポートを使用しなくなったときMIBClose API 9 0 4 呼び出しを行いポートを閉じる。

#### 【0051】

上位モジュールがMIBオブジェクトの読み出しを行う場合は、MIBReadObjects API 9 0 2 呼び出しを行う。MIBReadObjects API 9 0 2 呼び出しにはポート識別子、読み出すべきMIBオブジェクトのオブジェクトキーを指定すると共に、MIBモジュールが読み出したMIBオブジェクトの値を上位へ通知するためのコールバック関数のアドレスを指定する。

#### 【0052】

MIBReadObjects API 9 0 2 呼び出しによりSNMPのGet-request コマンドが生成され、ネットワーク上に送信される。図8に示したように、このGet-request コマンドに応答するエージェントを持つデバイスはGet-response コマンドを送信する。

#### 【0053】

上位モジュールがMIBオブジェクトへの書き込みを行う場合は、MIBWriteObjects API 9 0 3 呼び出しを行う。MIBWriteObjects API 9 0 3 呼び出しにはポー

ト識別子、書き込むMIBオブジェクトのオブジェクトキーとその値を指定すると共に、MIBモジュールが書き込みの結果を上位へ通知するためのコールバック関数のアドレスを指定する。

#### 【0054】

MIBWriteObjects API 9 0 3 呼び出しによりSNMPのGet-request コマンドが生成され、ネットワーク上に送信される。図8に示したように、このGet-request コマンドに応答するエージェントを持つデバイスはGet-response コマンドを送信する。

#### 【0055】

コールバック関数はMIBReadObjects API 9 0 2 もしくはMIBWriteObjects API 9 0 3 の結果を上位モジュールに通知するためのものである。具体的には、デバイスのアドレスと受信したGet-response コマンドの内容を上位へ通知する。

#### 【0056】

ブロードキャストアドレスを指定してオープンされたポートに対してMIBReadObjects API 9 0 2 を呼び出しを行った場合、ネットワーク上に送信されるGet-request コマンドを運ぶパケット（IPプロトコルの場合はIPパケット、NetWareプロトコルの場合はIPXパケット）の宛先アドレスがブロードキャストアドレスになる。従って、このパケットは複数のデバイスで受信されるので、Get-request コマンドには複数のデバイスが応答する。つまりマネージャ側では複数のGet-response コマンドを受信する。この場合、コールバック関数は、ポート識別子は同じでデバイスのアドレスが異なる複数回の呼び出しが行われる。上位モジュールはアドレス情報を調べることで、そのコールバックがどのデバイスからのものかを知ることができる。

#### 【0057】

次に具体的なデータの流れを説明する。MIBモジュール610では上位からの要求によりオブジェクトキーからオブジェクトIDへの変換等の処理を行いSNMPモジュール611へコマンド送信要求を行う。SNMPモジュール611はMIBモジュール610からの送信要求によりSNMP PDUをRAM 503上で組

み立て、共通トランスポートモジュール 6 1 2 へ送信要求を行う。共通トランスポートモジュール 6 1 2 では動作プロトコルによりヘッダの付加等の所定の処理を行い、TCP/IP プロトコルであれば WinSock モジュール 6 1 7 へ、NetWare プロトコルであれば NetWare API モジュール 6 1 6 へパケット送信要求を行う。以下 TCP/IP プロトコルで動作しているものとして説明を行う。WinSock モジュール 6 1 7 は送信要求のあったパケットを IP パケット化し、OS に対してネットワークへのデータ送信要求を行う。OS は RAM 5 0 3 上のデータをシステムバス 5 0 4 を介して NIC 5 0 8 へ書き込む。NIC 5 0 8 では書き込まれたデータを所定のフレーム化して LAN 1 0 0 に送信する。

## 【 0 0 5 8 】

LAN 1 0 0 に接続されているデバイスからのパケットは NIC 5 0 8 で受信される。NIC 5 0 8 ではパケット受信を割り込みにより OS に通知する。OS は NIC 5 0 8 から受信パケットをシステムバス 5 0 4 経由で読み出し RAM 5 0 3 に置く。OS では動作プロトコルもしくは受信したパケットからプロトコルを判断し、TCP/IP プロトコルであれば WinSock モジュール 6 1 7 へ、NetWare プロトコルであれば NetWare API モジュール 6 1 6 へパケット受信が通知される。以下 TCP/IP プロトコルで動作しているものとして説明を行う。WinSock モジュール 6 1 7 では、受信パケットが自分宛のものかどうかを受信パケット中のアドレスにより判断する。受信パケットが自分宛のものでは無いときは受信パケットを破棄する。受信パケットが自分宛であった場合は、UDP ハンドラ 6 1 4 起動し、共通トランスポートモジュール 6 1 2 にパケット受信を通知する。共通トランスポートモジュール 6 1 2 ではトランスポートヘッダの除去等の所定の処理を行い、SNMP モジュール 6 1 1 へパケット受信を通知する。SNMP モジュール 6 1 1 では SNMP ヘッダの除去等の所定の処理を行い、MIB モジュール 6 1 1 へ PDU 受信を通知する。MIB モジュール 6 1 0 では所定処理を行うと共に受信した情報を MIB API で規定された形式に変換し上位モジュールのコールバック関数を呼び出すことにより、デバイスからの応答を上位モジュールへ通知する。

## 【 0 0 5 9 】

なお、以下の説明において、本願に係るネットワーク管理ソフトウェアのことを「NetSpot」と呼称する。

【0060】

NetSpotのインストールに必要なファイルは、通常、フロッピーディスク（FD）やCD-ROMなどの物理媒体に記録されて配布されるか、あるいはネットワークを経由して伝送される。ユーザは、これらの手段によりNetSpotのインストールに必要なファイルを入手した後、所定のインストール手順に従ってNetSpotのインストールを開始する。

【0061】

NetSpotのインストール手順は、他の一般的なソフトウェアのインストール手順と同様である。すなわち、ユーザがNetSpotのインストーラをパーソナルコンピュータ（PC）上で起動すると、その後はインストーラが自動的にインストールを実行する。インストーラは、NetSpotの動作に必要なファイルをPCのハードディスクにコピーし、また、必要に応じてユーザから情報を入力してもらいながら、NetSpotの動作に必要なファイルの修正または新規作成なども行う。

【0062】

次に、本従来例におけるネットワーク管理プログラムにおける探索シーケンスについて説明する。

【0063】

図10は、従来のネットワーク管理プログラムにおける探索シーケンスを示す図である。

【0064】

図10において、探索モジュール1030はネットワーク管理プログラムの探索モジュールを示し、これは図6における604と同等である。この探索モジュールは、ネットワーク管理プログラムの他のモジュールと同様に、図1におけるPC103上で図5におけるCPU501によって実行される。

【0065】

デバイス1031は、ネットワーク上に接続されており、かつSNMPエージェントが動作しているデバイスでプリンタとして標準的なMIBのみを搭載して

いるプリンタを示し、例えば図 1 における NB 1 1 8 を示す。ここで、標準的な M I B とは、R F C 1 2 1 3、R F C 1 5 1 4 および R F C 1 7 5 9 に規定されている M I B などと言う。

【 0 0 6 6 】

デバイス 1 0 3 2 は、ネットワーク上に接続されており、かつ S N M P エージェントが動作しているデバイスで、プリンタとして標準的な M I B および企業独自のプライベート M I B を搭載しているプリンタを示し、例えば図 1 における NB 1 0 1 を示す。本実施例では、NB 1 0 1 が企業独自のプライベート M I B を実装しているものとして説明を行なう。この企業独自のプライベート M I B は、図 4 のノード 4 0 8 で示される c a n o n ノード以下の M I B オブジェクトである。

【 0 0 6 7 】

上位モジュールから探索開始の指示が出されると、探索モジュールはブロードキャストアドレスを指定してデバイスの状態とデバイス種別を取得するための G e t リクエスト S N M P パケットをネットワークに送出する（1 0 0 1 および 1 0 0 2）。このパケットは、ネットワークに接続されている全てのデバイスに届けられる。

【 0 0 6 8 】

ここで、問い合わせる M I B オブジェクトとして、1 0 0 1 では標準的な M I B のみを、1 0 0 2 ではプライベート M I B を用いる。異なる M I B オブジェクトを問い合わせるのは、プライベート M I B を搭載しているデバイスと、標準 M I B のみを搭載しているデバイスの両方を見つけないためである。

【 0 0 6 9 】

この S N M P パケットに対して、S N M P エージェントを実装しているネットワークデバイスは、それぞれ応答パケットを送出する（1 0 0 3 から 1 0 0 6）。

【 0 0 7 0 】

ここで 1 0 0 3 は、標準 M I B を問い合わせるブロードキャストパケット 1 0 0 1 に対するデバイス 1 0 3 1 の応答であり、デバイス 1 0 3 1 の状態に応じた



値が返る。1004は、標準MIBを問い合わせるブロードキャストパケット1001に対するデバイス1032の応答であり、デバイス1032の状態に応じた値が返る。1005は、プライベートMIBを問い合わせるブロードキャストパケット1002に対するデバイス1031の応答であり、デバイス1031はプライベートMIBを実装していないため、エラーとしてnoSuchNameが返される。1006は、プライベートMIBを問い合わせるブロードキャストパケット1002に対するデバイス1032の応答であり、デバイス1032の状態に応じた値が返る。

#### 【0071】

探索を開始してから一定の時間が経過すると一時応答タイマー1023が満了するので、探索モジュールはさらに詳細な情報を取得すべく、それぞれのデバイスに対してSNMPパケットを送信する。

#### 【0072】

より具体的には、一時探索タイマー1023が満了した時点までに、デバイス1031からはプライベートMIBに対してnoSuchNameエラーが返ってきたことがわかっているので、このデバイスをプライベートMIBには対応していないデバイス、すなわち標準MIBのみに対応しているデバイスとみなして、標準MIBを用いてより詳細な取得を行なう。この取得およびデバイス1031の返答が1007および1008である。

#### 【0073】

同様に、デバイス1032からは標準MIBおよびプライベートMIBの両方から正常な応答が返ってきているので、このデバイスをプライベートMIBを実装しているデバイスとみなして、プライベートMIBを用いてより詳細な取得を行なう。

#### 【0074】

この取得およびデバイス1032の返答が1009および1010である。さらに時間が経過してデバイス応答タイマー1021が満了した時点で、上位モジュールに対してそれまでに探索したデバイスの情報を通知する。

#### 【0075】

さらに時間が経過し、探索間隔タイマー 1 0 2 2 が満了した時点で、再度探索を開始する。以降今まで説明した動作と全く同一のシーケンスであるので、説明を省略する。

【0 0 7 6】

【発明が解決しようとする課題】

しかしながら上記従来例では、デバイス管理装置が能動的にブロードキャストパケットを送信することによりデバイスの探索を行なう。

【0 0 7 7】

このようにブロードキャストパケットを送信することは、そのパケットに対して SNMP に応答できる全てのネットワーク機器が返信パケットを送信することになり、ネットワークのトラフィックを著しく大きくしてしまう。定期的にブロードキャストパケットを送信することはブロードキャストパケットによるトラフィック増大現象が定期的に起きることを意味し、そのような状況はネットワークの負荷および管理の観点から好ましくない。ブロードキャストパケットの送信は、できる限り少ないことが望ましい。

【0 0 7 8】

本発明は上記従来例に鑑みてなされたもので、ネットワークの負荷を増大させずにネットワークデバイスの情報を取得することがでできるネットワークデバイス管理装置及び方法を提供することを目的とする。

【0 0 7 9】

【課題を解決するための手段】

上記目的を達成するために、本発明は次のような構成からなる。すなわち、起動後に少なくとも 1 度はネットワーク管理用パケットをブロードキャストするデバイスが接続されたネットワークの管理装置であって、

パケットを受信するパケット受信手段と、

前記パケット受信手段で受信したパケットが、ネットワーク管理用のパケットであるか判定するパケット判定手段と、

前記パケット判定手段でネットワーク管理用パケットであると判断された場合、そのパケットから、そのパケットを送信したデバイスのアドレスを取得するデ

バースアドレス取得手段と、

前記バースアドレスを登録するバースアドレス登録手段とを備える。

【0080】

あるいは、起動後に少なくとも1度はネットワーク管理用パケットをブロードキャストするデバイスが接続されたネットワークの管理装置であって、

パケットを受信するパケット受信手段と、

前記パケット受信手段で受信したパケットが、ネットワーク管理用のパケットであるか判定するパケット判定手段と、

前記パケット判定手段でネットワーク管理用パケットであると判断された場合、そのパケットを送信したデバイスに対して、そのデバイスが監視の対象となるデバイスであることを確認するための確認パケットを送信する送信手段と、

前記監視パケットに対する応答に基づいて、その応答を返したデバイスの状態をアドレスと共に登録するバースアドレス登録手段とを備える。

【0081】

また好ましくは、前記応答は、その応答元のデバイスが管理対象のデバイスであれば正常応答であり、該デバイスの稼働状態を示す情報を含む。

【0082】

また好ましくは、前記送信手段は、プリンタの状態を取得するための確認パケットを送信し、前記アドレス登録手段は、確認パケットに対する応答が正常応答であれば、応答元のデバイスのアドレスと共に、そのデバイスがプリンタであることを登録する。

【0083】

【発明の実施の形態】

(第1の実施形態)

従来例の説明で述べたように、SNMPの規定には、トラップと呼ばれるパケット構造がある。トラップにはいくつかの種類があるが、それらのうちコールドスタートトラップと呼ばれるものは、機器が起動した時にネットワーク上にブロードキャストされる。本発明では、このコールドスタートトラップを捉えることにより、ネットワークに新しいデバイスが追加されたことを検知しようとするも

のである。

【0084】

以下、図面を用いて詳細に説明する。

【0085】

図11は、本実施例において、各種の機器のネットワーク上での配置を示す図である。

【0086】

1101は、ネットワークデバイス管理プログラムが実行されるコンピュータ、1102はディレクトリサービスが動作しているディレクトリサーバ、1103はトラップパケットを監視するトラップ監視プログラムを実行するトラップ監視プログラム実行マシンである。1101、1102および1103のコンピュータの構成は、従来例の説明において図5で示したものと同様の構成が使える。ただし、ディレクトリサーバ1102およびトラップ監視プロセス実行マシン1103においては、キーボードコントローラ505、キーボード509、CRTコントローラ506、CRT510のように、ユーザとの対話を行なうための入出力関連の部分は必ずしも必要ではない。さらに、1101、1102および1103は、同一のコンピュータであっても良い。

【0087】

なお、上記で言うディレクトリサービスとは、言わばネットワークに関する電話帳であり、様々な情報を格納するためのものである。ディレクトリシステム的具体例としては、例えばLDAP (Lightweight Directory Access Protocol)がある。LDAPの規定は、IETFが発行しているRFC1777に記載されている。また解説書としては、例えば株式会社プレンティスホールより「LDAPインターネット ディレクトリ アプリケーション プログラミング」が1997年11月1日に発行されている。

【0088】

本実施例では、ディレクトリサーバとしてLDAPサーバを用いることにして説明を行なうが、LDAPサーバ以外のディレクトリサーバでも本発明の本質を損なわない。

## 【 0 0 8 9 】

図 1 2 は、トラップ監視プログラム実行マシン 1 1 0 3 で動作する、トラップの監視プログラムの動作を示すフローチャートである。トラップ監視プログラムは、トラップ監視プログラム実行マシン 1 1 0 3 が起動すると同時に起動するように設定されているか、あるいはネットワークの管理者によって明示的に起動されるものとする。

## 【 0 0 9 0 】

トラップ監視プログラムでは、まずステップ S 1 2 0 1 でディレクトリサーバ 1 1 0 2 にネットワークを介して接続し、情報を登録する準備をする。接続するディレクトリサーバのアドレスは、システム管理者によってあらかじめ登録されており、例えばファイルに記録されてハードディスク 5 1 1 に格納されているものとする。次にステップ S 1 2 0 2 で、パケットの受信があったかどうかを調べる。これは、OS が提供する API を呼び出すことにより、OS が NIC 5 0 8 の状態を調べることによって行われる。パケットの受信があった場合ステップ S 1 2 0 3 に進み、受信パケットが SNMP トラップであるかどうかを判断する。受信パケットのフォーマットが図 7 で説明した SNMP パケットに合致しており、かつ PDU タイプ 7 1 1 のフィールドの値がトラップを示す値であったときに SNMP トラップパケットであると判断する。ステップ S 1 2 0 3 で受信パケットが SNMP トラップであると判断された場合にはステップ S 1 2 0 4 に進む。ステップ S 1 2 0 4 では、受信パケットの送信元アドレスフィールドを見て、送信したデバイスのアドレスを取り出す。次にステップ S 1 2 0 5 に進み、ステップ S 1 2 0 4 で取り出したデバイスのアドレスをディレクトリサーバ 1 1 0 2 に送信してディレクトリサービスに登録する。

## 【 0 0 9 1 】

一方、ステップ S 1 2 0 3 で SNMP トラップ受信ではないと判断された場合、ステップ S 1 2 0 6 に進み SNMP トラップ受信以外の受信処理を行なう。あるいは、ステップ S 1 2 0 2 でパケット受信ではないと判断された場合には、ステップ S 1 2 0 7 に進み、パケット受信以外のその他の処理を行なう。

## 【 0 0 9 2 】

ステップ S 1 2 0 5、S 1 2 0 6 あるいは S 1 2 0 7 の処理が終了したら、ステップ S 1 2 0 2 に戻って処理を続ける。

【 0 0 9 3 】

図 1 3 は、クライアント 1 1 0 1 で実行されるデバイス制御プログラムにおいて、デバイスリストを表示する動作を示すフローチャートである。図 1 3 の手順は、クライアント 1 1 0 1 においてデバイス管理プログラムが起動した場合、あるいはユーザの操作によって明示的に起動される。

【 0 0 9 4 】

デバイスリスト表示動作においては、まずステップ S 1 3 0 1 で、ディレクトリサーバに接続する。接続するディレクトリサーバのアドレスは、システム管理者によってあらかじめ登録されており、例えばファイルに記録されてハードディスク 5 1 1 に格納されているものとする。そのアドレスは、トラップ監視プログラム実行マシン 1 1 0 3 のトラップ監視プログラムが接続するディレクトリサーバのアドレスと同じである。次にステップ S 1 3 0 2 において、ステップ S 1 3 0 1 で接続したディレクトリサーバから、ディレクトリサービスに登録されている情報を取り出す。次にステップ S 1 3 0 3 で、C R T C 5 0 6 を操作して C R T 5 1 0 上にステップ S 1 3 0 2 で取得した情報を表示する。

【 0 0 9 5 】

図 1 4 は、ステップ S 1 3 0 3 における表示の様子を示す図である。ウィンドウ 1 4 0 1 に、ディレクトリサーバから取得したデバイスのアドレス一覧が表示されている。

【 0 0 9 6 】

以上のようにして、トラップ監視プロセスにより、ネットワーク上のデバイスが起動時にブロードキャストするコールドスタートトラップを補足してそのデバイスのアドレスを取得し、それをディレクトリサーバに登録する。こうすることで、デバイスの探索を能動的に行うことなくネットワークに接続されているデバイスのアドレスを把握できる。このため、能動的に探索パケットの発行が不要となり、ネットワークのトラフィックが軽減できる。

【 0 0 9 7 】

## (第 2 の実施の形態)

トラップパケットは、SNMPによるデバイス管理において一般的なものであるので、トラップパケットが、探索の対象としているデバイス種別からのものであることを確認してからディレクトリサービスに登録した方が、後にクライアントがディレクトリサービスにアクセスすることを考えると都合がよい。つまり、例えばクライアントがネットワークプリンタのみを制御対象にしているような場合には、トラップパケットを送信したデバイスがプリンタであることを確認してからディレクトリサービスに登録する方がより望ましい。

## 【0098】

さらにトラップパケットを受信した時には、デバイスの種別を確認するためのパケットとしてデバイスの状態を問い合わせるパケットを用いることにより、デバイスに関するより詳細な情報を得ることができる。

## 【0099】

本発明第 2 の実施例は、これらの点の改良を図ったものである。

## 【0100】

以下第 2 の実施例においては、制御対象のネットワークデバイスがネットワークプリンタであるとして説明を進める。

## 【0101】

図 1 5 は、本発明の第 2 の実施例において、トラップ監視プログラム実行マシン 1 1 0 3 で実行されるトラップ監視プログラムの動作について説明したフローチャートである。

## 【0102】

この図においてステップ S 1 5 0 1 から S 1 5 0 4 までは、第 1 の実施例において図 1 2 を用いて説明した S 1 2 0 1 から S 1 2 0 4 までの動作と全く同じであるので説明を省略する。同様に、S 1 5 0 8 と S 1 2 0 6 および S 1 5 0 9 と S 1 2 0 7 はそれぞれ同じであるので説明を省略する。

## 【0103】

ステップ S 1 5 0 5 においては、トラップパケットを送信してきたデバイスがネットワークプリンタであることを確認するために、ネットワークプリンタがい

ンプリメントしているであろうMIBオブジェクトを取得するためのSNMPパケットを送信する。より具体的には、例えばRFC1514 Host Resources MIBに規定されているhrPrinterDetectedErrorStateを取得すべくGet-Nextパケットを送信する。このMIBオブジェクトは、プリンタの状態を表すものであり、デバイスに問い合わせる情報としてこのMIBオブジェクトを用いることにより、プリンタであることを確認するとともに、同時にデバイスに関するより詳細な情報を得ることができる。

#### 【0104】

ステップS1503でSNMPトラップ以外のパケットの受信であると判断されたら、ステップS1506に進み、そのパケットがSNMPパケットで、かつネットワークプリンタを示す正常応答であるかどうかを判断する。より具体的には、受信したパケットがステップS1505で送信したGet-Nextパケットに対するGet-Response応答パケットであることをリクエストIDフィールド712の値が同じであることを確認することにより行なうとともに、MIB情報フィールド715に含まれている取得したMIBのオブジェクトIDがhrPrinterDetectedErrorState.Xであることを確認する。ここでXは、トラップパケットを送信したデバイスのMIBエージェントにおけるプリンタデバイスを示すインデックス値を示す。

#### 【0105】

もし取得したMIBのオブジェクトIDがhrPrinterDetectedErrorState.Xであった場合には、トラップパケットを送信したデバイスをネットワークプリンタであると判断し、ステップS1507に進む。一方、オブジェクトIDがhrPrinterDetectedErrorStateの次のオブジェクトであった場合、あるいはエラーステータスフィールド713の値がオブジェクトなし(noSuchObject)であった場合には、トラップパケットを送信したデバイスはプリンタではないと判断し、ステップS1508に進む。

#### 【0106】

ステップS1507では、ステップS1501で接続したLDAPサーバに、トラップパケットを送信したデバイスのネットワークアドレス、そのデバイスが



プリンタであることを示すフラグおよびステップ S 1 5 0 6 で取得した hrPrinterDetectedErrorState.X の値を登録する。

【0 1 0 7】

図 1 6 は、ステップ S 1 5 0 7 で登録した情報をクライアント 1 1 0 1 が表示している様子を示した図である。表示手順については、図 1 3 で説明したのと同様であるので省略する。

【0 1 0 8】

図 1 3 の情報と比較して、デバイスがプリンタであることがアイコンで示されている他、デバイスの状態に関する情報が追加されている。例えば 1 6 0 3 は、デバイスに何らかの警告エラーが生じていることを示している。1 6 0 2 は、デバイスタイプがプリンタであるとの確認がとれていないデバイスを示している。

【0 1 0 9】

以上のようにして、トラップ監視プロセスにより、ネットワーク上のデバイスが起動時にブロードキャストするコールドスタートトラップを補足してそのデバイスのアドレスを取得し、そのアドレスを有するデバイスに対して、その状態を問い合わせる。この問合せに対する応答に応じて、アドレスと共に得られた状態をディレクトリサーバに登録する。こうすることで、デバイスの探索を能動的に行うことなくネットワークに接続されているデバイスのアドレスを把握できる。このため、能動的に探索パケットの発行が不要となり、ネットワークのトラフィックが軽減できる。それに加えて、状態の問合せにより、そのデバイスの種類や稼働状態をディレクトリに登録することができ、利用者はより詳細な情報を獲得することができる。

【0 1 1 0】

上記で説明した本発明に係るネットワークデバイス探索プログラムは、外部からインストールされるプログラムによって、P C 5 0 0 によって遂行されても良い。その場合、そのプログラムは C D - R O M やフラッシュメモリやフロッピーディスクなどの記憶媒体により、あるいは電子メールやパソコン通信などのネットワークを介して、外部の記憶媒体からプログラムを含む情報群を P C 5 0 0 上にロードすることにより、P C 5 0 0 に供給される場合でも本発明は適用される

ものである。

【0 1 1 1】

図 1 7 は、記憶媒体の一例である C D - R O M のメモリマップを示す図である。9 9 9 9 はディレクトリ情報を記憶してある領域で、以降のインストールプログラムを記憶してある領域 9 9 9 8 およびネットワークデバイス探索プログラムを記憶してある領域 9 9 9 7 の位置を示している。9 9 9 8 は、インストールプログラムを記憶してある領域である。9 9 9 7 は、ネットワークデバイス制御プログラムを記憶してある領域である。本発明のネットワーク探索プログラムが P C 5 0 0 にインストールされる際には、まずインストールプログラムを記憶してある領域 9 9 9 8 に記憶されているインストールプログラムがシステムにロードされ、C P U 5 0 1 によって実行される。次に、C P U 5 0 1 によって実行されるインストールプログラムが、ネットワークデバイス探索プログラムを記憶してある領域 9 9 9 7 からネットワークデバイス探索プログラムを読み出して、ハードディスク 5 1 1 に格納する。

【0 1 1 2】

なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェース機器、リーダなど）から構成されるシステムあるいは統合装置に適用しても、ひとつの機器からなる装置に適用してもよい。

【0 1 1 3】

また、前述した実施形態の機能を実現する、図 1 2、図 1 3 あるいは図 1 5 に記載された手順のプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（または C P U や M P U ）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、本発明の目的が達成される。

【0 1 1 4】

この場合、記憶媒体から読み出されたプログラムコード自体が本発明の新規な機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0 1 1 5】

プログラムコードを供給するための記憶媒体としては、例えば、フロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモ리카ード、ROMなどを用いることができる。

【0 1 1 6】

また、コンピュータが議み出したプログラムコードを実行することによって、前述した実施形態の機能が実現される他、そのプログラムコードの指示に基づき、コンピュータ上で稼動しているOSなどが実際の処理の一部または全部を行い、その処理によっても前述した実施形態の機能が実現され得る。

【0 1 1 7】

さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によっても前述した実施形態の機能が実現され得る。

【0 1 1 8】

なお、本発明は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体から、そのプログラムをパソコン通信など通信ラインを介して要求者にそのプログラムを配信する場合にも適用できる。

【0 1 1 9】

【発明の効果】

以上説明したように、本発明によれば、デバイス管理装置が能動的にブロードキャストパケットを送信することなくネットワークデバイスの情報を取得することができる。このため、デバイス探索のためにネットワークの負荷が増大することを防止できる。

【図面の簡単な説明】

【図 1】

プリンタをネットワークに接続するためのネットワークボードを、開放型アーキテクチャを持つプリンターへつなげた場合を示す図である。

【図 2】

エージェントを実装したネットワークボードをプリンタに接続する実施形態を示す断面図である。

【図 3】

ネットワークボードとプリンターと LAN との電氣的接続を示すブロック図である。

【図 4】

M I B の構造を示す概念図である。

【図 5】

ネットワーク管理ソフトウェアが稼動可能な P C の構成を示すブロック図である。

【図 6】

ネットワーク管理ソフトウェアのモジュール構成図である。

【図 7】

S N M P メッセージのフォーマットである。

【図 8】

マネージャとエージェント間での S N M P コマンドのやりとりを示す図である。

【図 9】

M I B モジュール A P I を示す図である。

【図 1 0】

従来例における探索モジュールとデバイス間の通信シーケンスを示す図である。

【図 1 1】

機器のネットワーク上での配置を示す図である。

【図 1 2】

トラップの監視プログラムの動作を示すフローチャートである。

【図 1 3】

デバイスリストを表示する動作を示すフローチャートである。

【図 1 4】

デバイス一覧の表示の様子を示す図である。

【図 1 5】

第 2 の実施例におけるトラップの監視プログラムの動作を示すフローチャートである。

【図 1 6】

第 2 の実施例におけるデバイス一覧の表示の様子を示す図である。

【図 1 7】

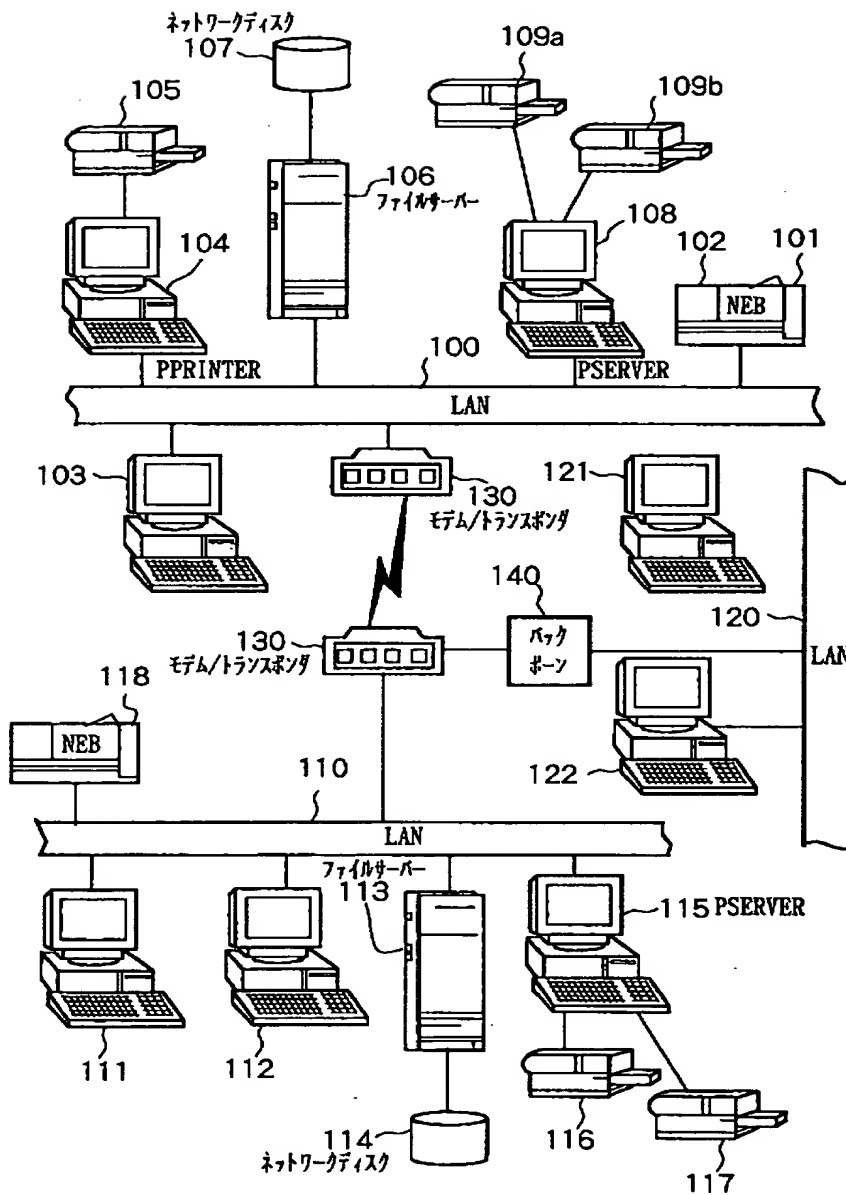
本発明のネットワーク管理ソフトウェアの記憶媒体におけるメモリマップを示す図である。

【符号の説明】

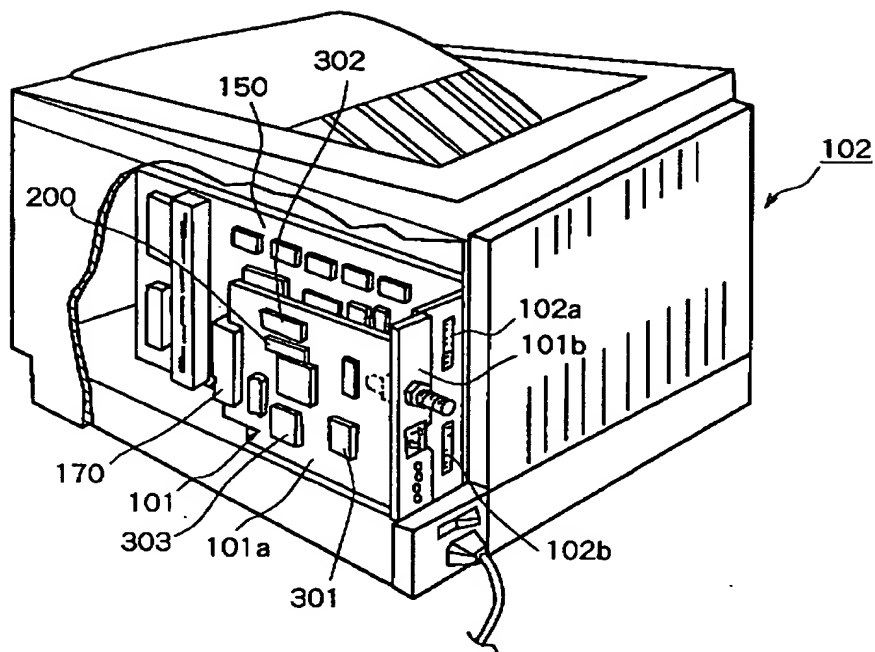
- S 1 1 0 1 クライアント
- S 1 1 0 2 ディレトリサーバ
- S 1 1 0 3 トラップ監視プログラム実行マシン
- S 1 1 0 4 ネットワークデバイス
- S 1 2 0 2 パケット受信手段
- S 1 2 0 3 トラップパケット受信判断手段
- S 1 2 0 4 デバイスアドレス取得手段
- S 1 2 0 5 デバイスアドレス登録手段
- S 1 5 0 5 確認パケット送信手段
- S 1 5 0 6 応答パケット確認手段

【書類名】 図面

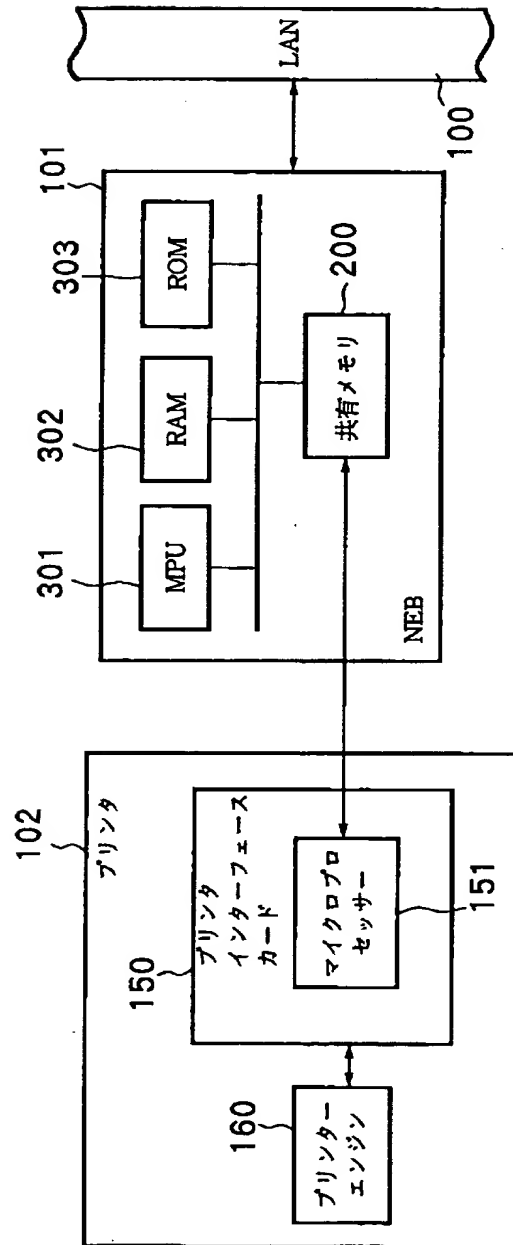
【図 1】



【図 2】

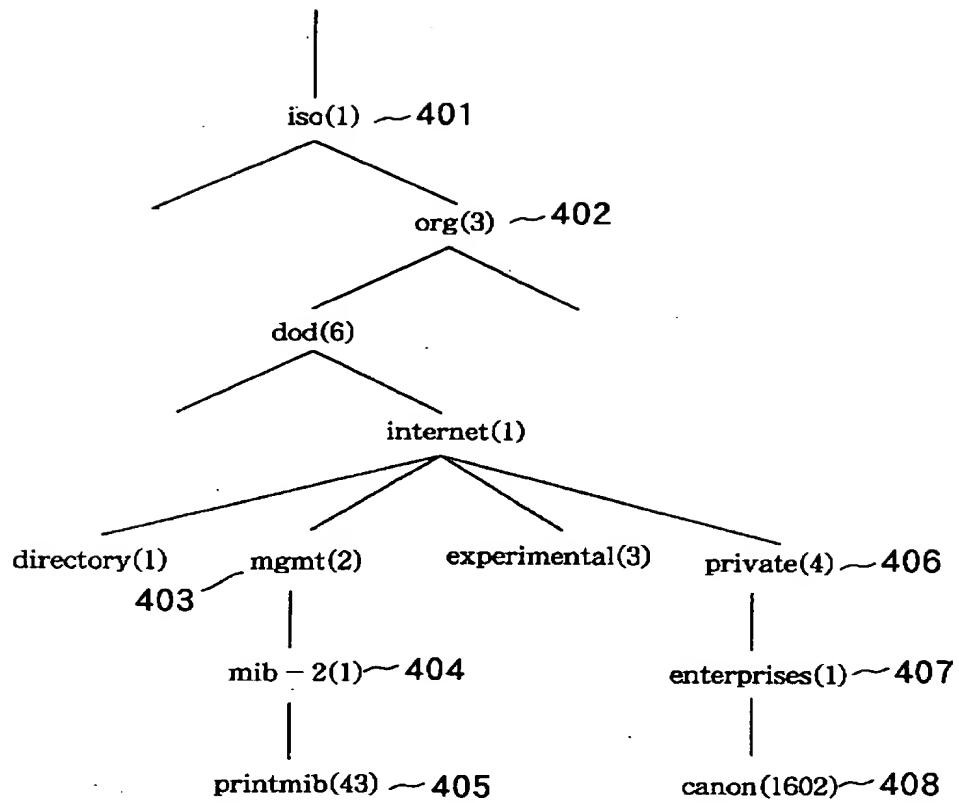


【図 3】

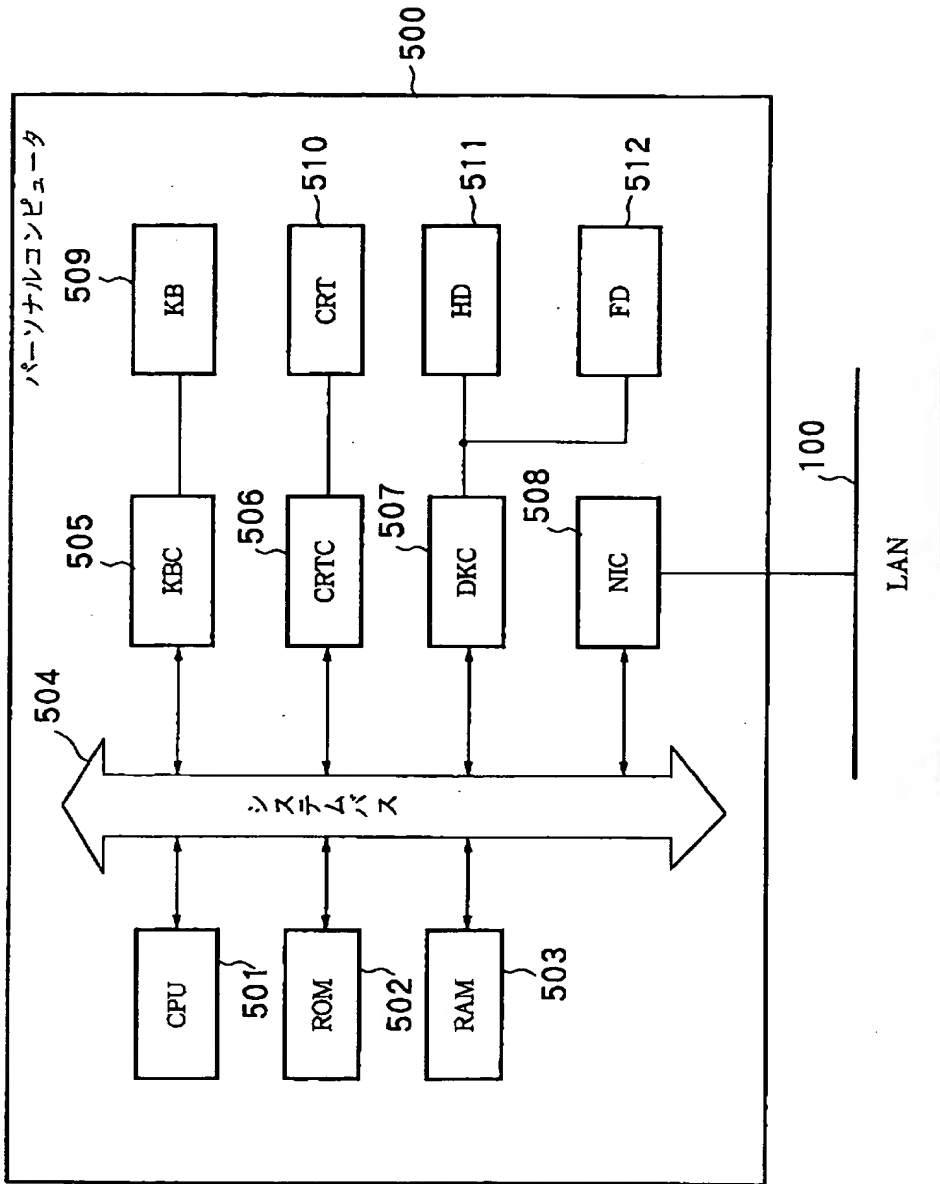




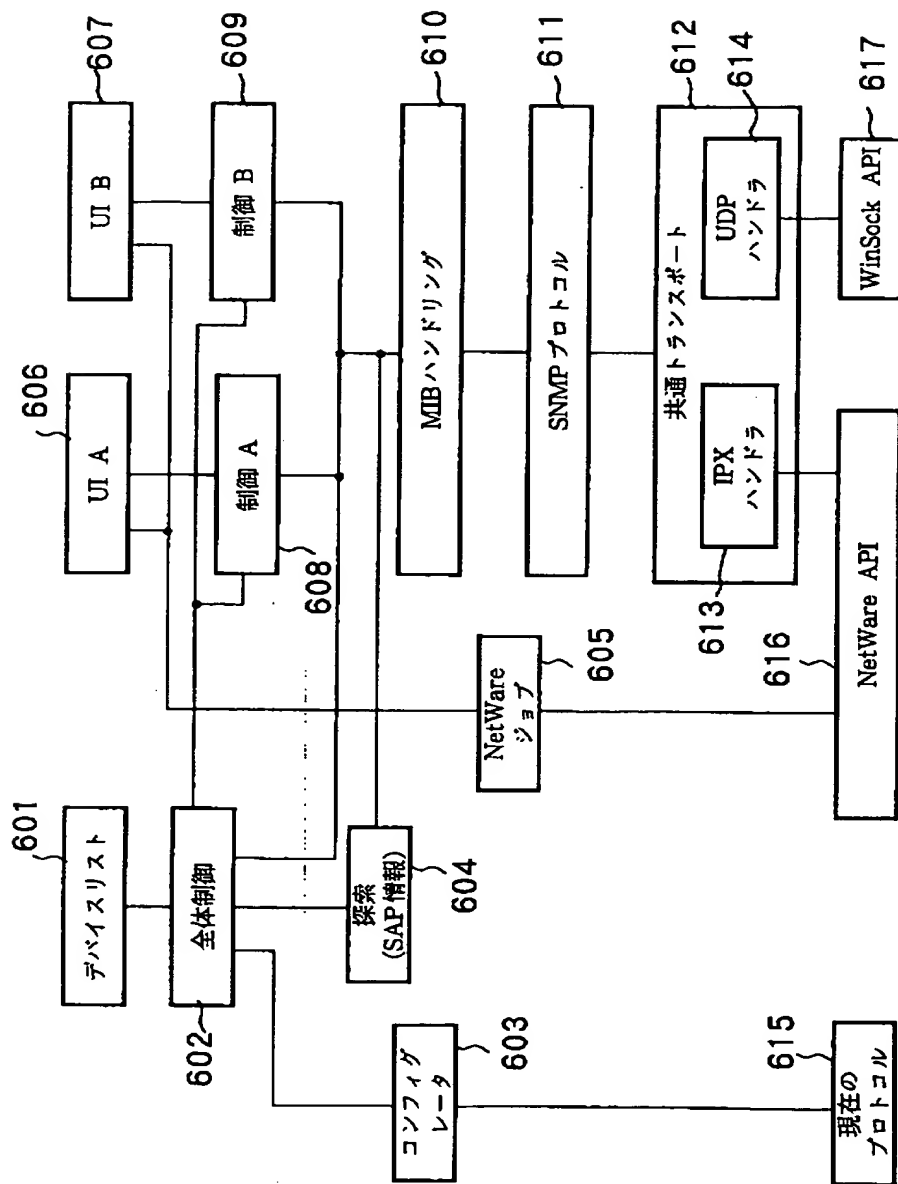
【図 4】



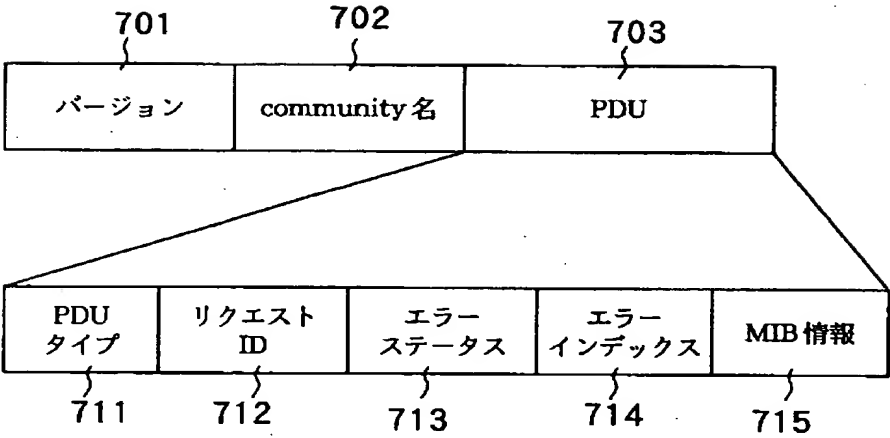
【図 5】



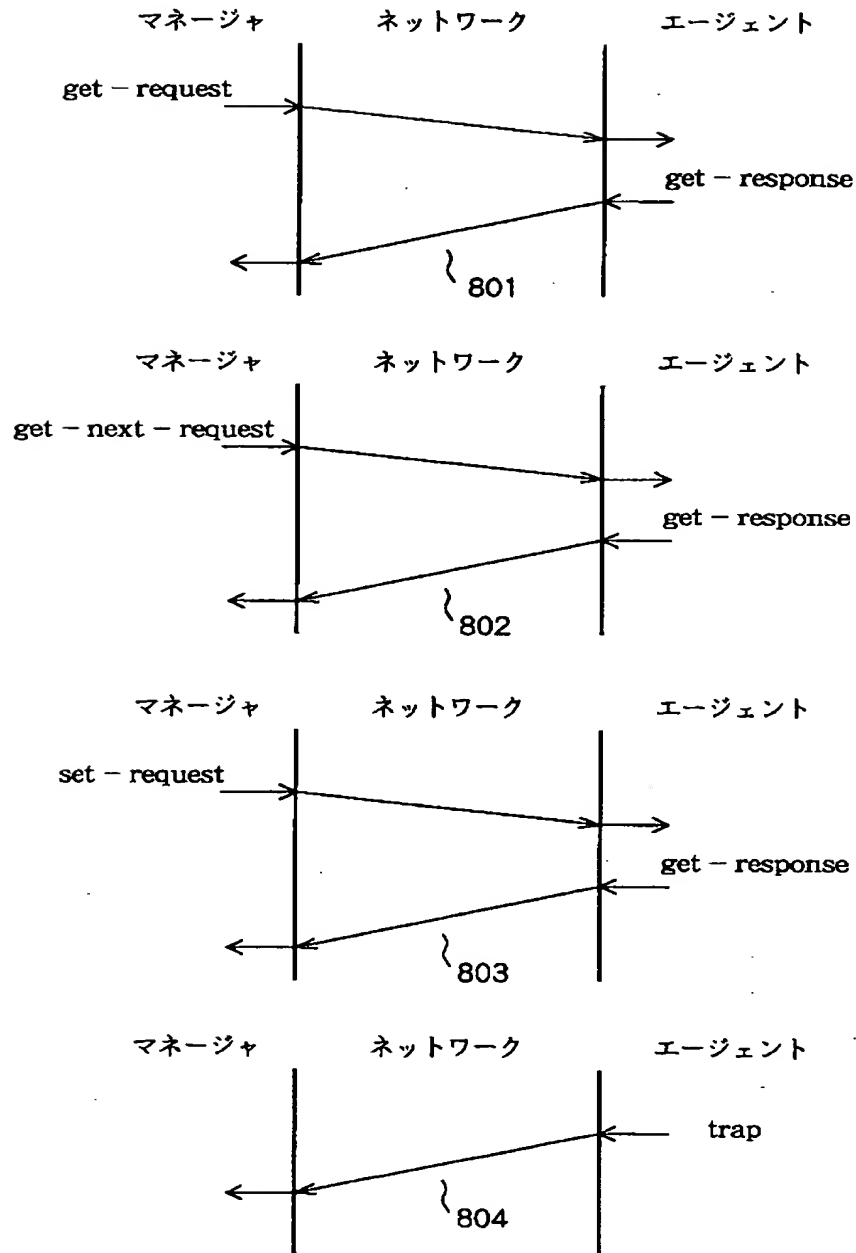
【図 6】



【図 7】



【図 8】



【図 9】

```

                                901
BOOL MIBOpen (int *port, ADDR ADDR *addr ) ;

                                902
BOOL MIBReadObjects (int port, int count, MIBOBJ *obj,
                                CALLBACK respproc ) ;

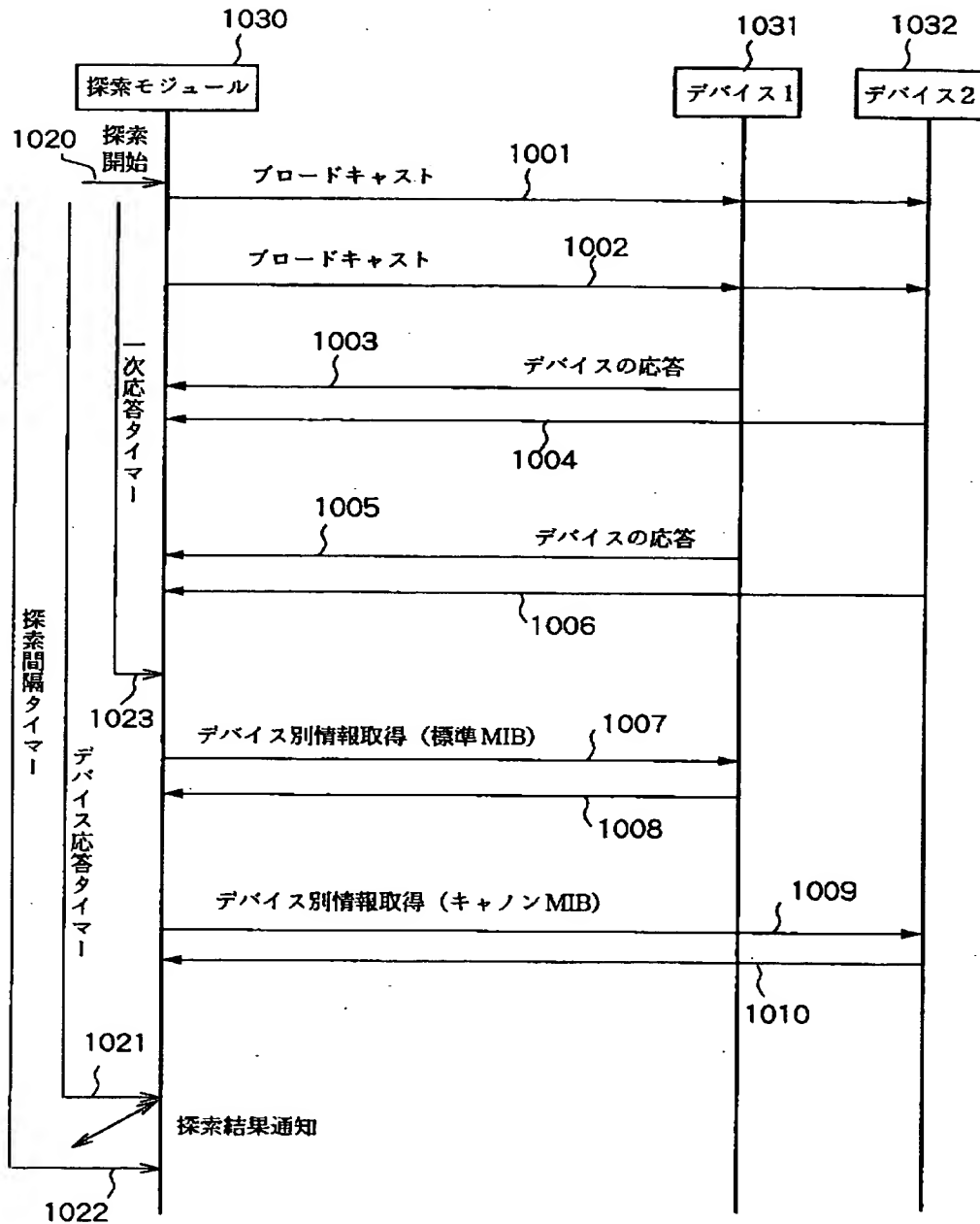
                                903
BOOL MIBWriteObjects (int port, int count, MIBOBJVAL *objval,
                                CALLBACK respproc ) ;

                                904
BOOL MIBClose( int port ) ;

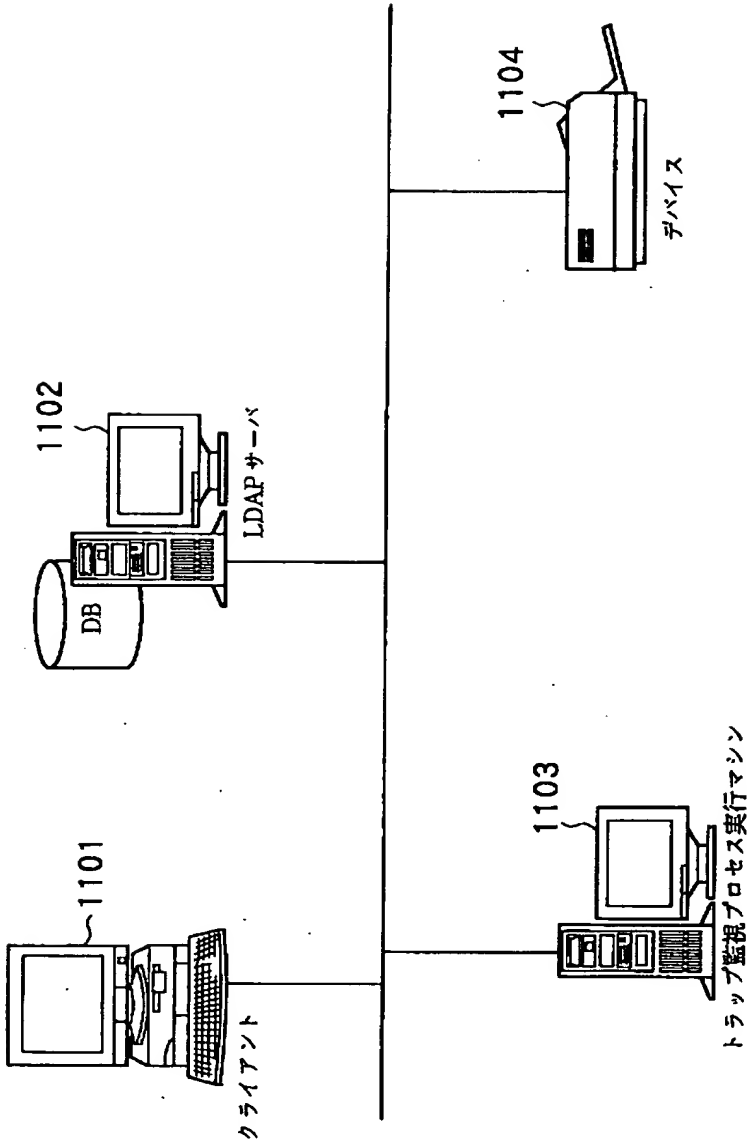
                                905
typedef VOID (*CALLBACK) (int port, ADDR *addr, INT result,
                                int count, MIBOBJVAL *objval ) ;

```

【図 1 0】

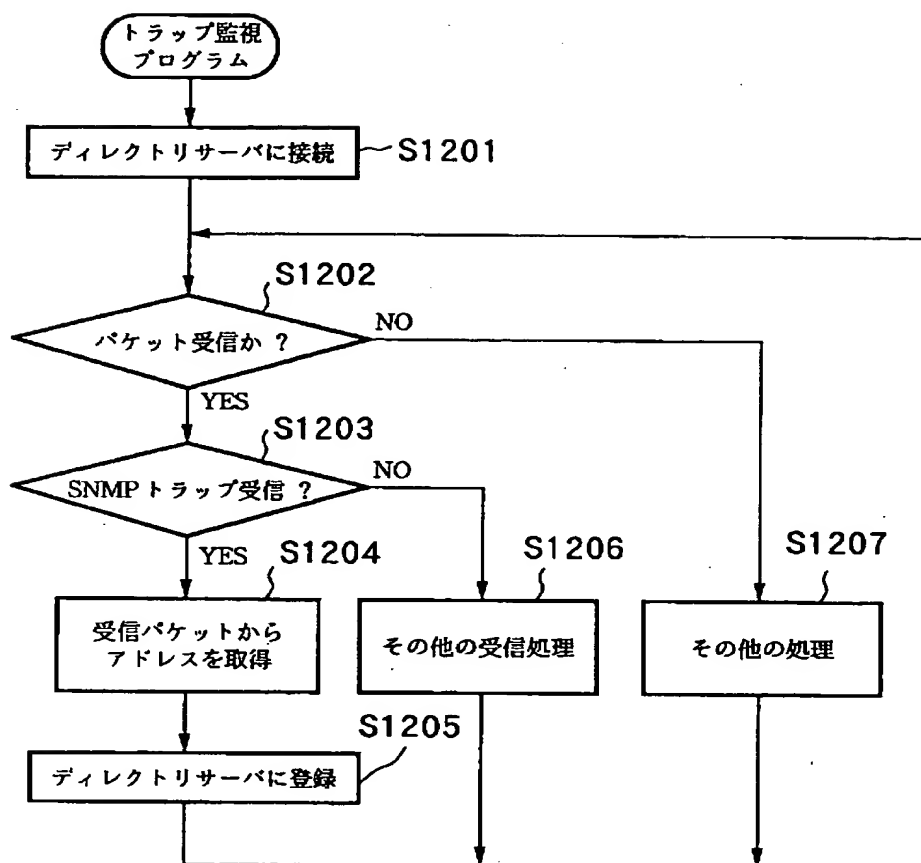


【図 1 1】

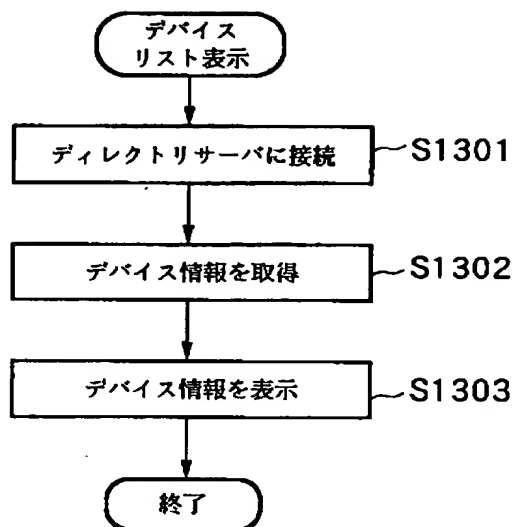




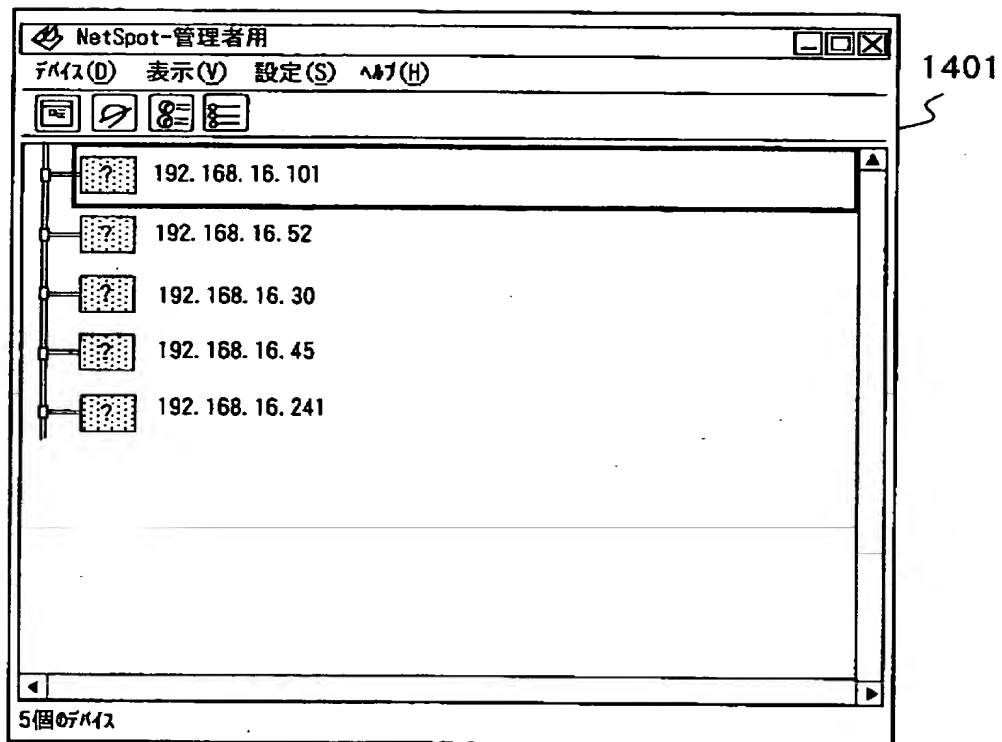
【図 1 2】



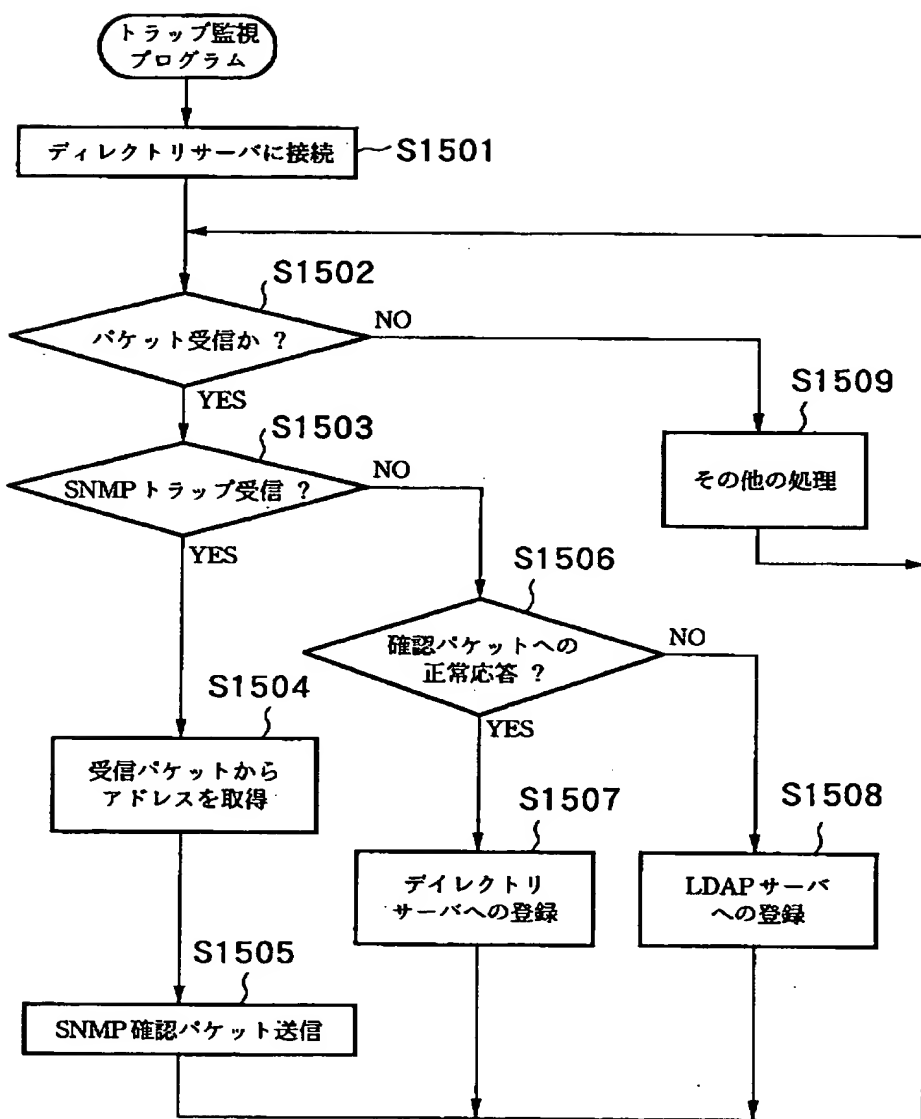
【図 1 3】



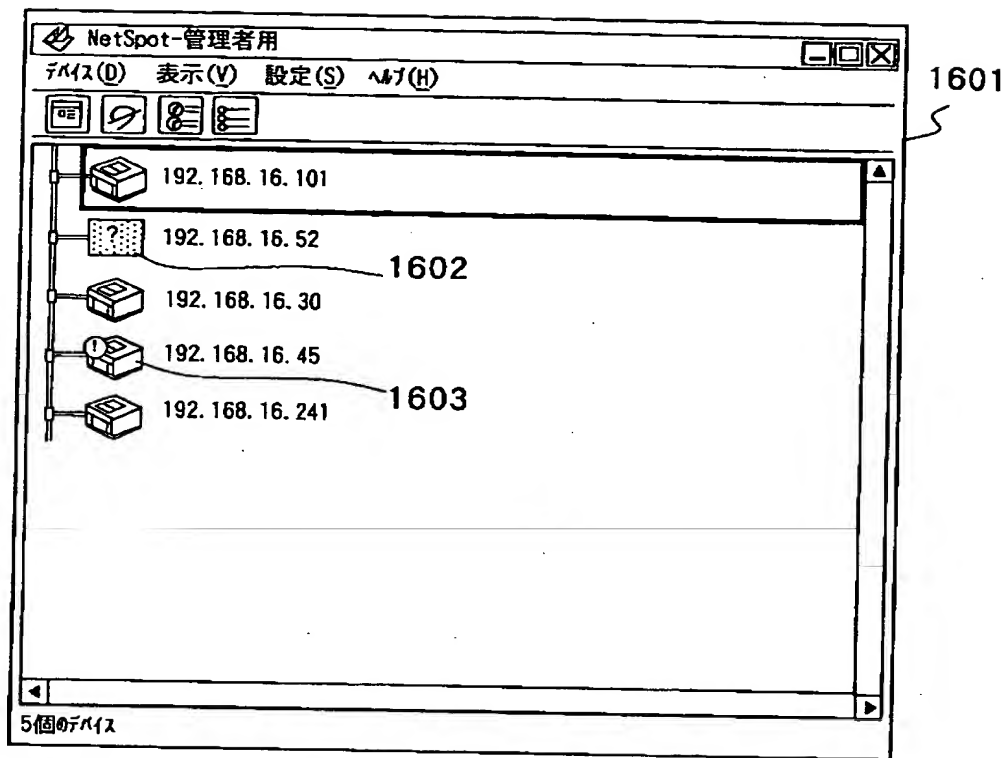
【図 1 4】



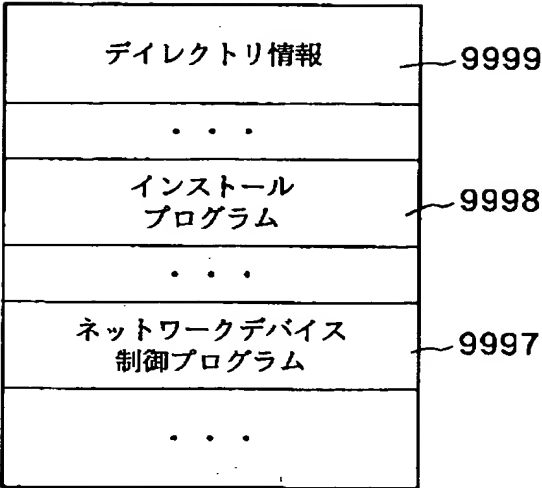
【図 15】



【図 1 6】



【図 1 7】



【書類名】 要約書

【要約】

【課題】 ネットワークデバイスの探索のためのトラフィックを減らす。

【解決手段】 各ネットワークデバイスは、起動時にコールドスタートトラップをブロードキャストする。監視装置はそれを補足し、SNMPパケットであるか判定して（S 1 2 0 3）、そうであれば受信パケットからアドレスを取り出してディレクトリサーバに登録する。

【選択図】 図 1 2

出 願 人 履 歴 情 報

識別番号 [000001007]

1. 変更年月日 1990年 8月30日  
[変更理由] 新規登録  
住 所 東京都大田区下丸子3丁目30番2号  
氏 名 キヤノン株式会社